



## Introduction to

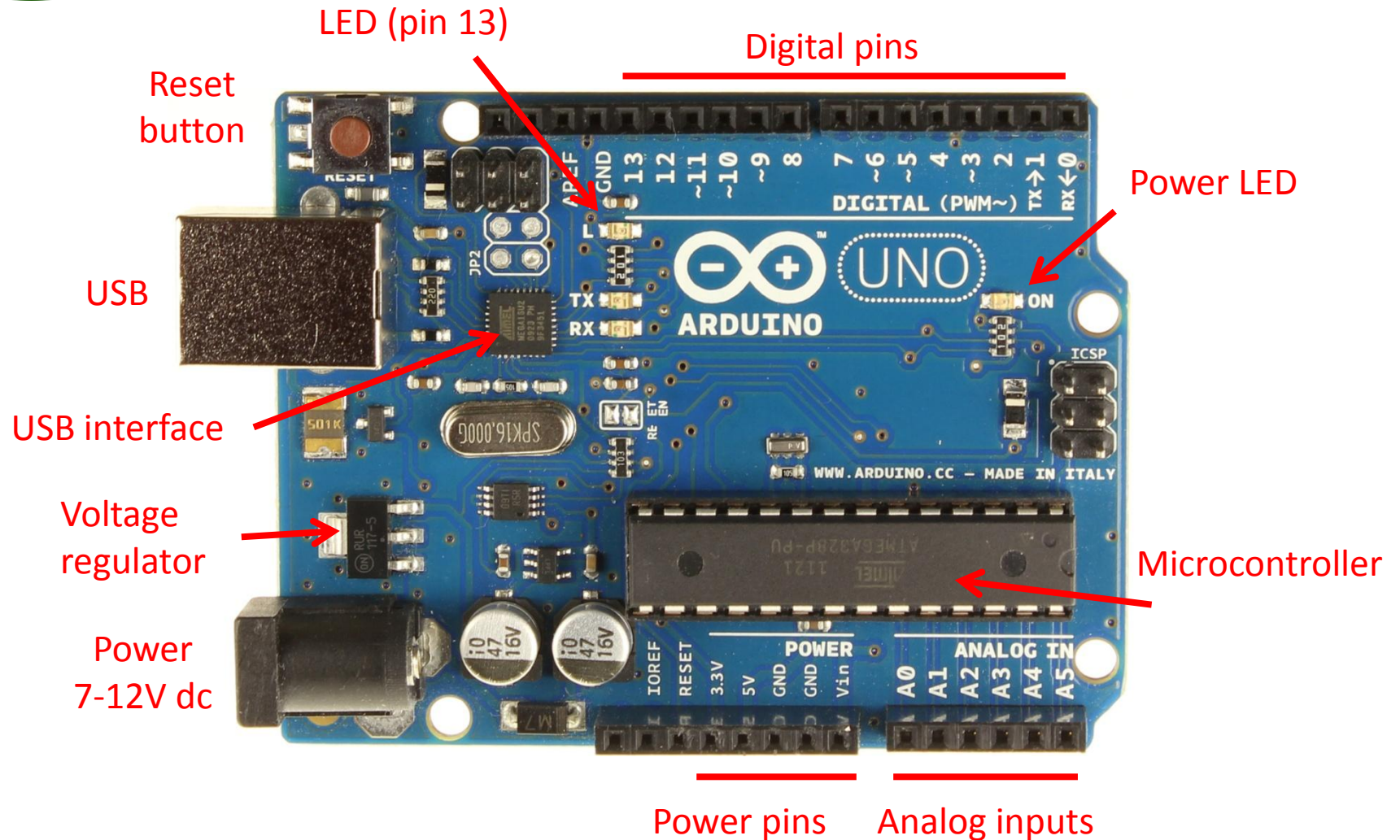
# ARDUINO

How to use electronics to make your projects better!

Presentation: <http://wiki.nottinghack.co.uk/wiki/Arduino101>  
Software: <http://arduino.cc/en/Main/Software>



# Arduino Uno





# Programming

- Arduino IDE  
(Integrated Development Environment)
- Program is called a “sketch”

The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The code is as follows:

```
Blink | Arduino 1.0.1
File Edit Sketch Tools Help

/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

At the bottom of the window, the hardware is identified as 'Arduino Nano w/ ATmega168 on COM8'.



# Blink a LED

You must have this.  
All initialisation  
goes here.

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
```

You must have this.  
The main code goes  
here.

```
void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```



# Serial monitor

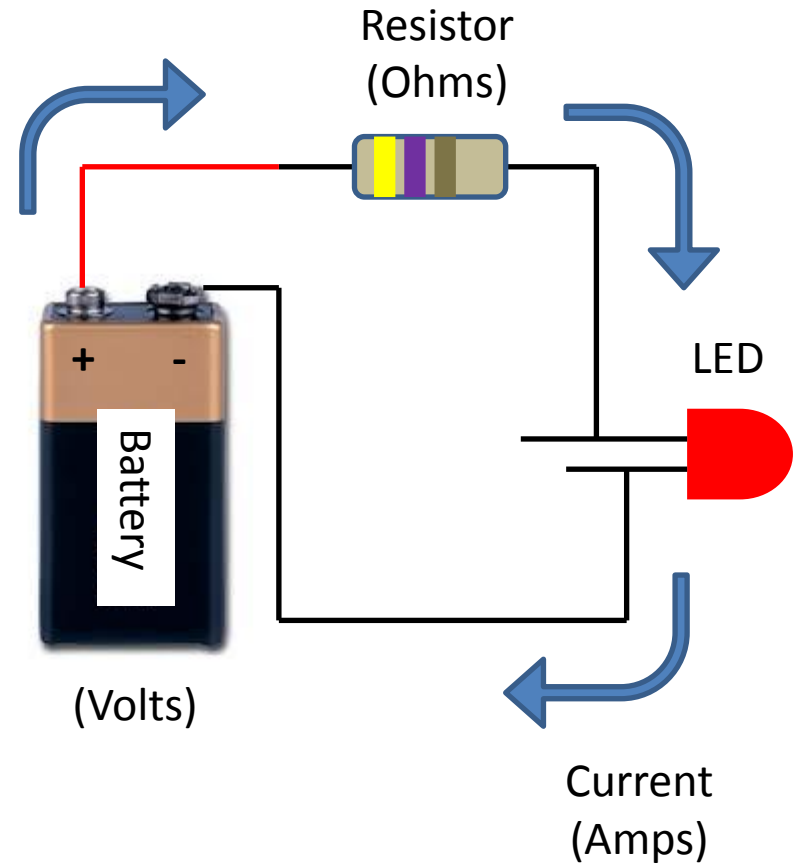
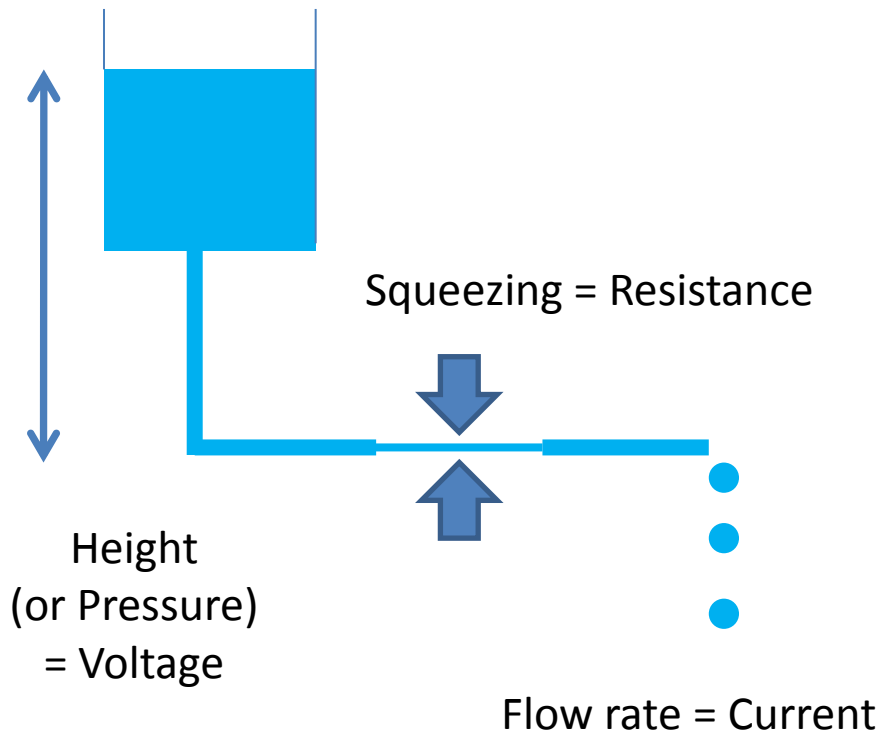
```
int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
  Serial.begin(9600);      // start the serial port at 9600 baud
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  Serial.println("LED on");
  delay(1000);              // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  Serial.println("LED off");
  delay(1000);              // waits for a second
}
```



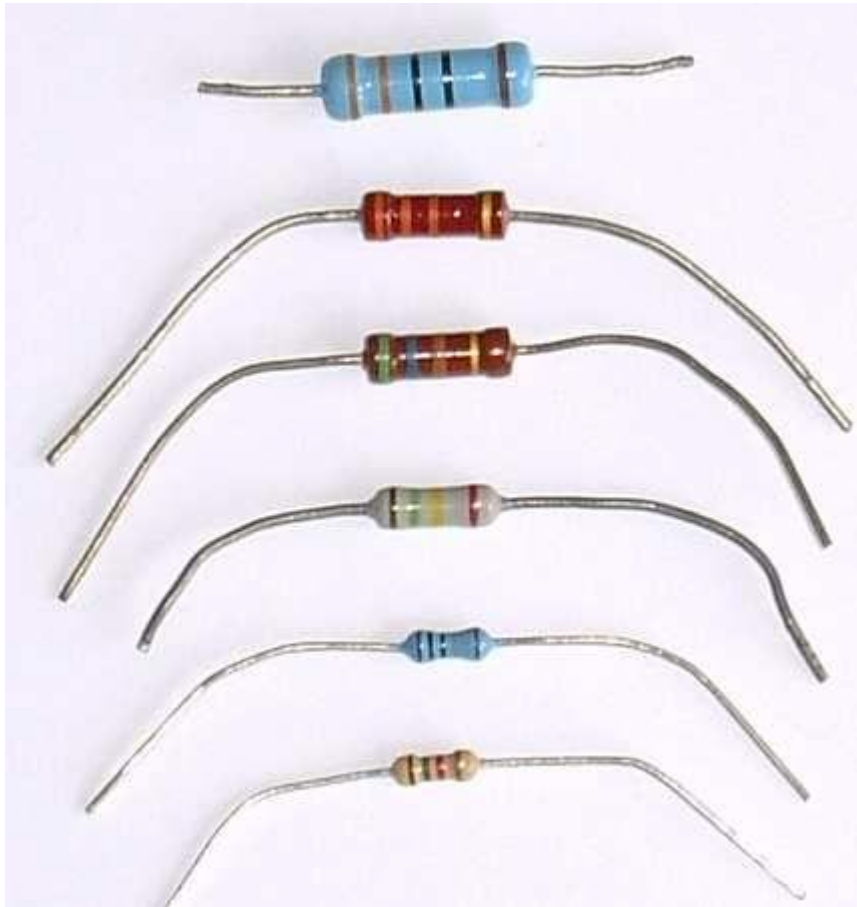
# Electrical concepts



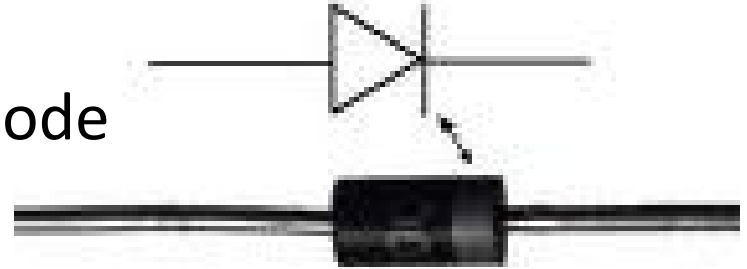


# Components

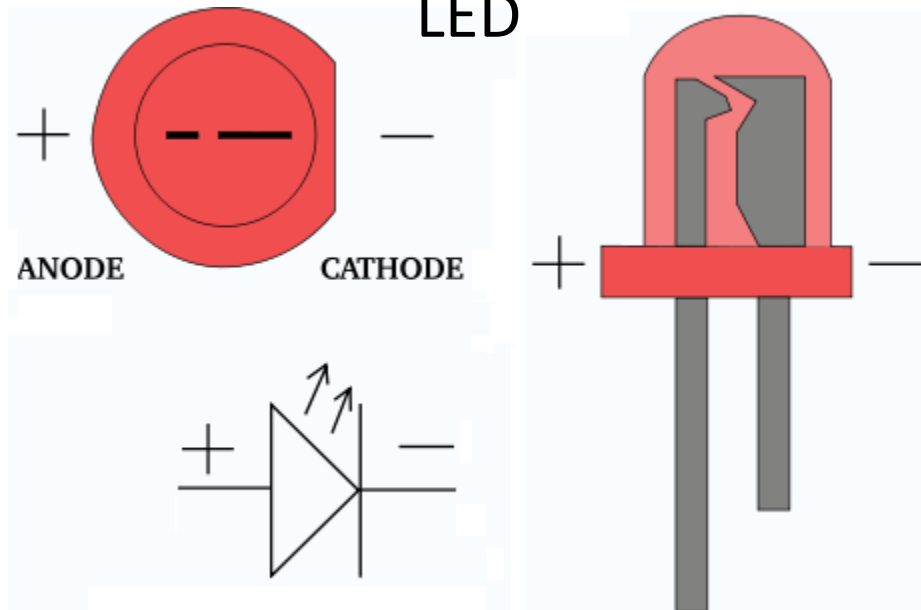
Resistor



Diode

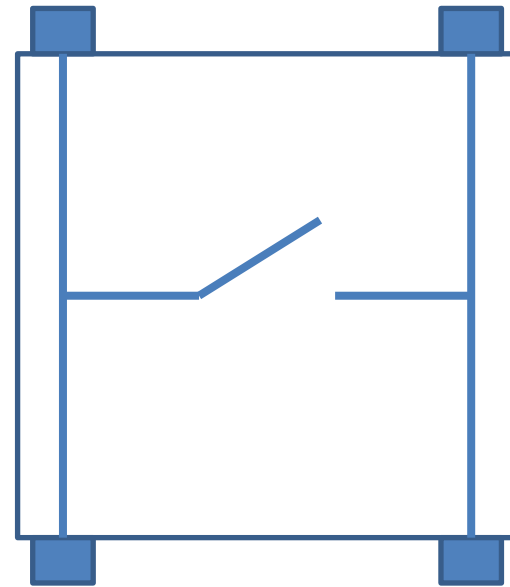
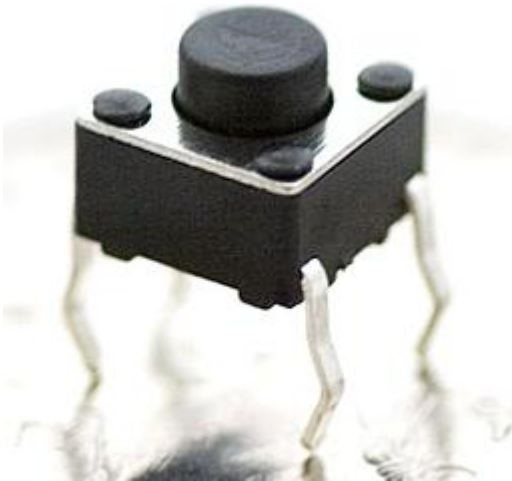


LED





# Switch

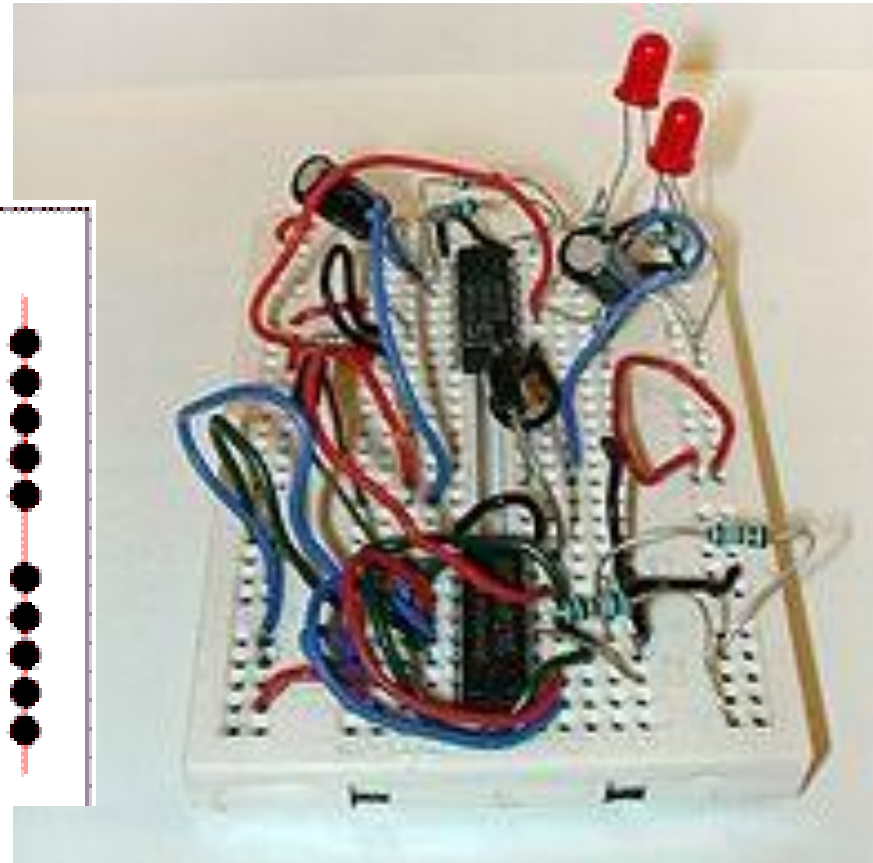
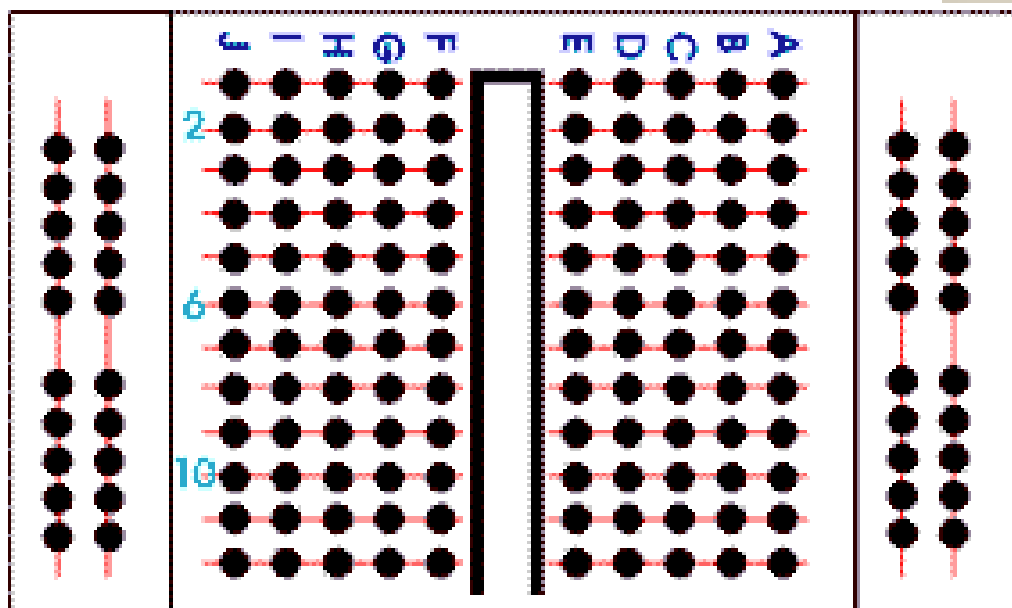






# Tools

## Breadboard





# Blink another LED

Alter the previous sketch:

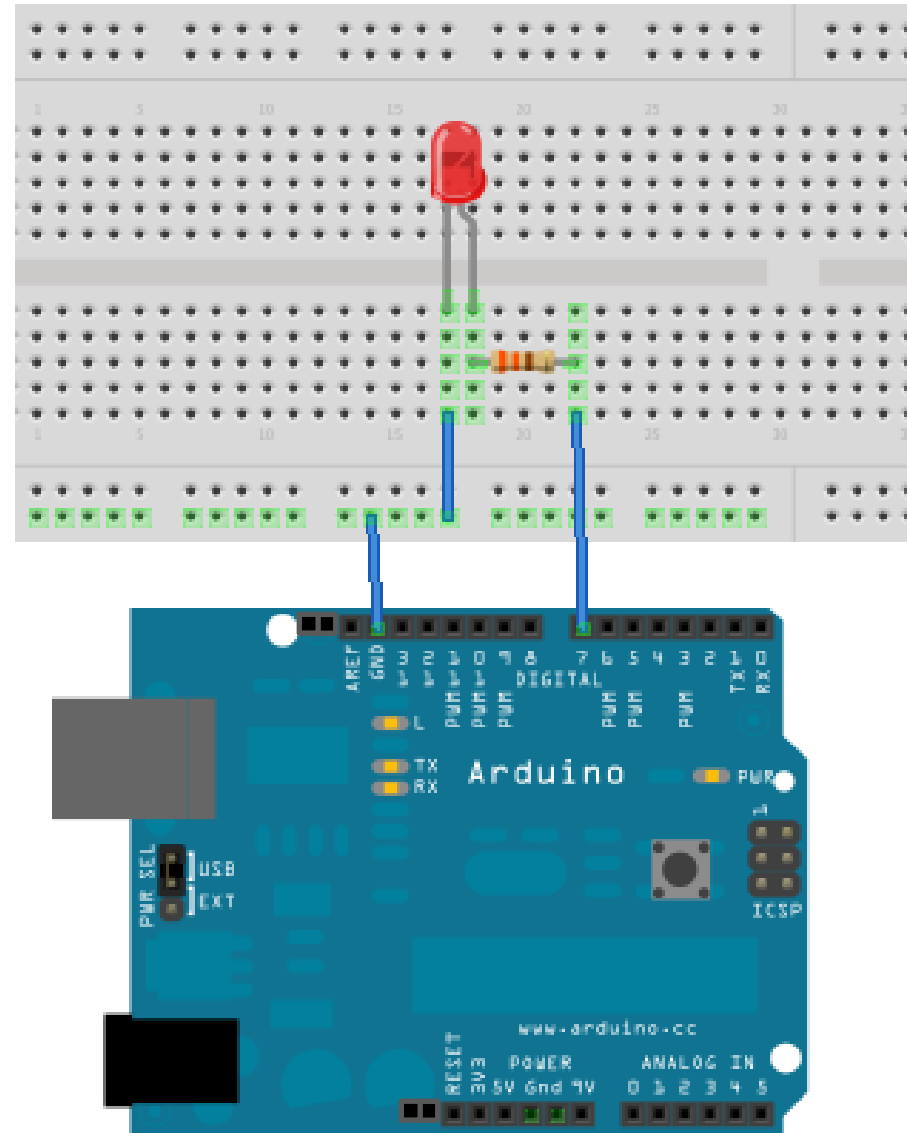
```
int ledPin = 13;
```

to

```
int ledPin = 7;
```

Exercises:

- Alter the blink frequency
- Two short blinks and one long blink
- Blink both LEDs alternately





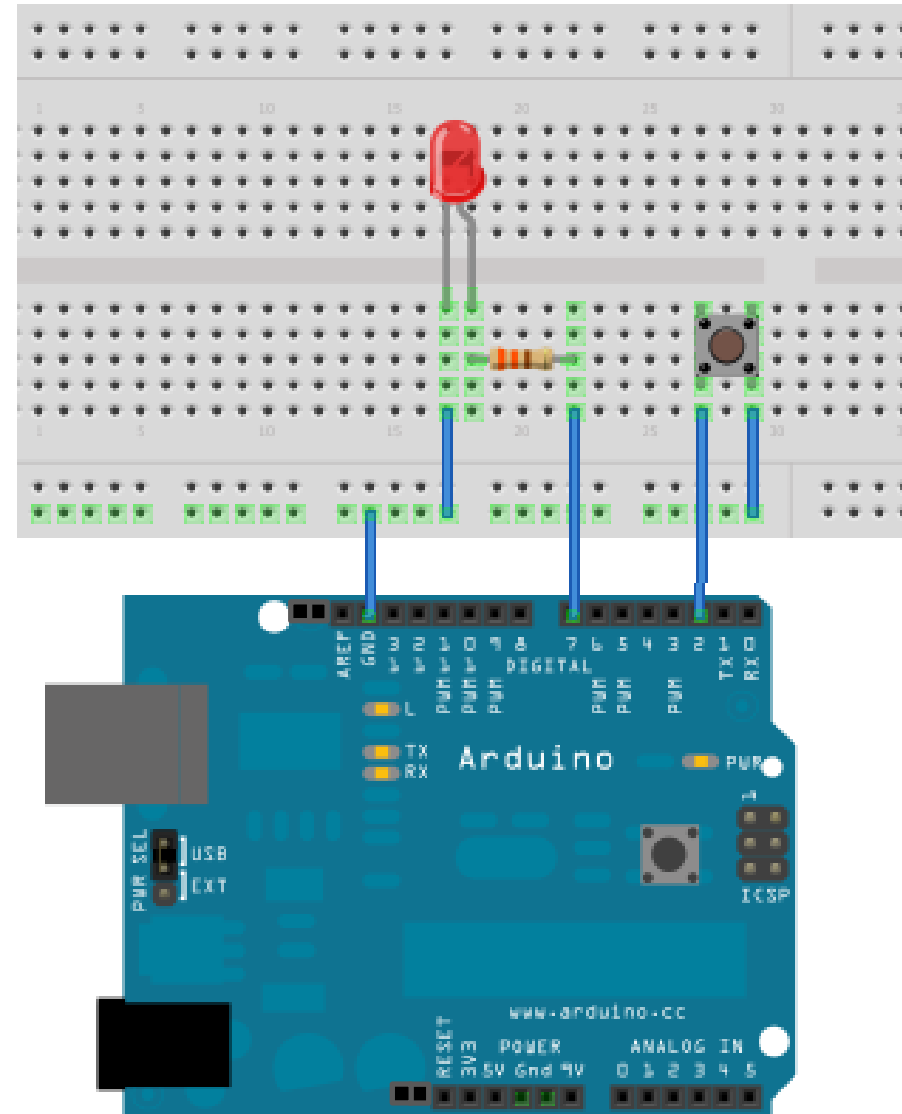
# Switch controlling LED

```
int ledPin = 13;
int switchPin = 2;

int switchvalue;    // store the switch value

void setup ()
{
  pinMode(ledPin, OUTPUT);
  // Set the switch pin as input with pull-up
  pinMode(switchPin, INPUT_PULLUP);
}

void loop ()
{
  // read the switch value
  switchvalue = digitalRead(switchPin);
  // and write it to the LED
  digitalWrite(ledPin, !switchvalue);
}
```





# Exercise

Blink the LED twice each time the switch is pressed.

Hint use an “if” statement

```
if (switchvalue == LOW)
{
    ... code to blink LED twice ...
}
```

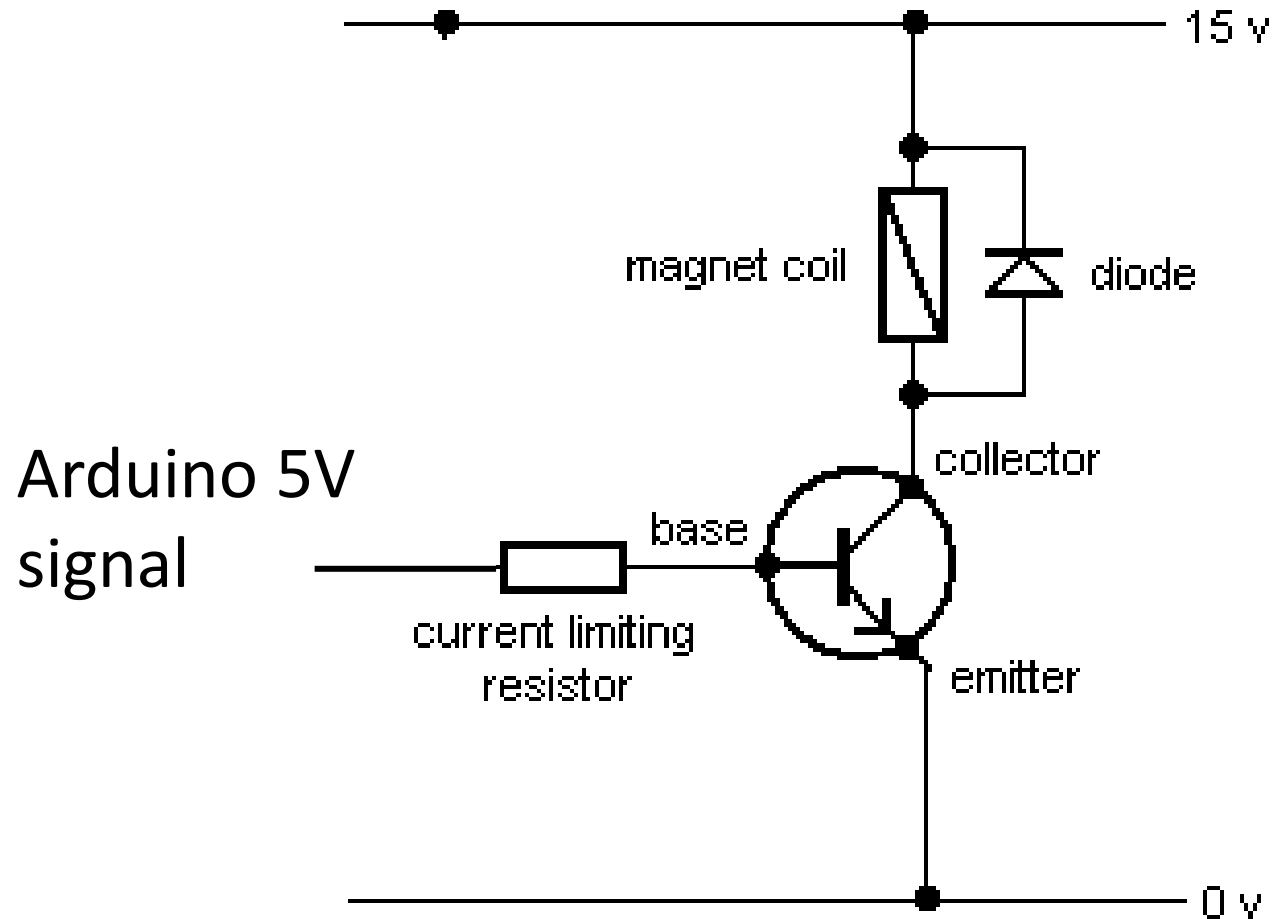


# Motors and relays

- These devices need more current than the Arduino can supply.
- We use a transistor to increase the current available.
- Transistor acts as a Arduino controlled switch.
- Need protection from inductive loads (usually wound coils such as motors and relays).
- NPN Bipolar and N-channel MOSFET are commonly used with the Arduino.
- Voltage of the load (motors, etc.) can be higher than Arduino voltage.



# NPN Bipolar transistor

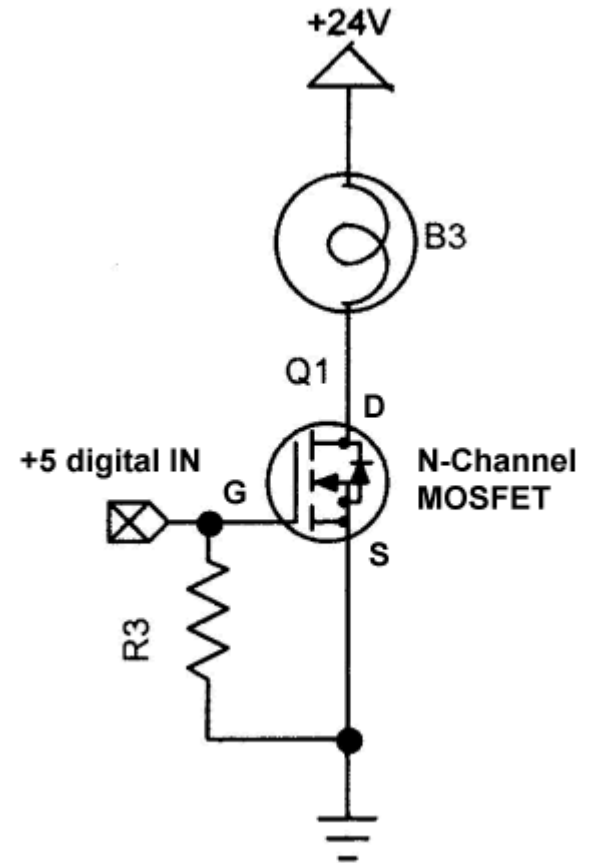




# N-channel MOSFET

Voltage controlled device

R3 ensures that the MOSFET turns off if the input is disconnected





# Speed control

## Pulse width modulation (PWM)

Drives motor at full voltage.

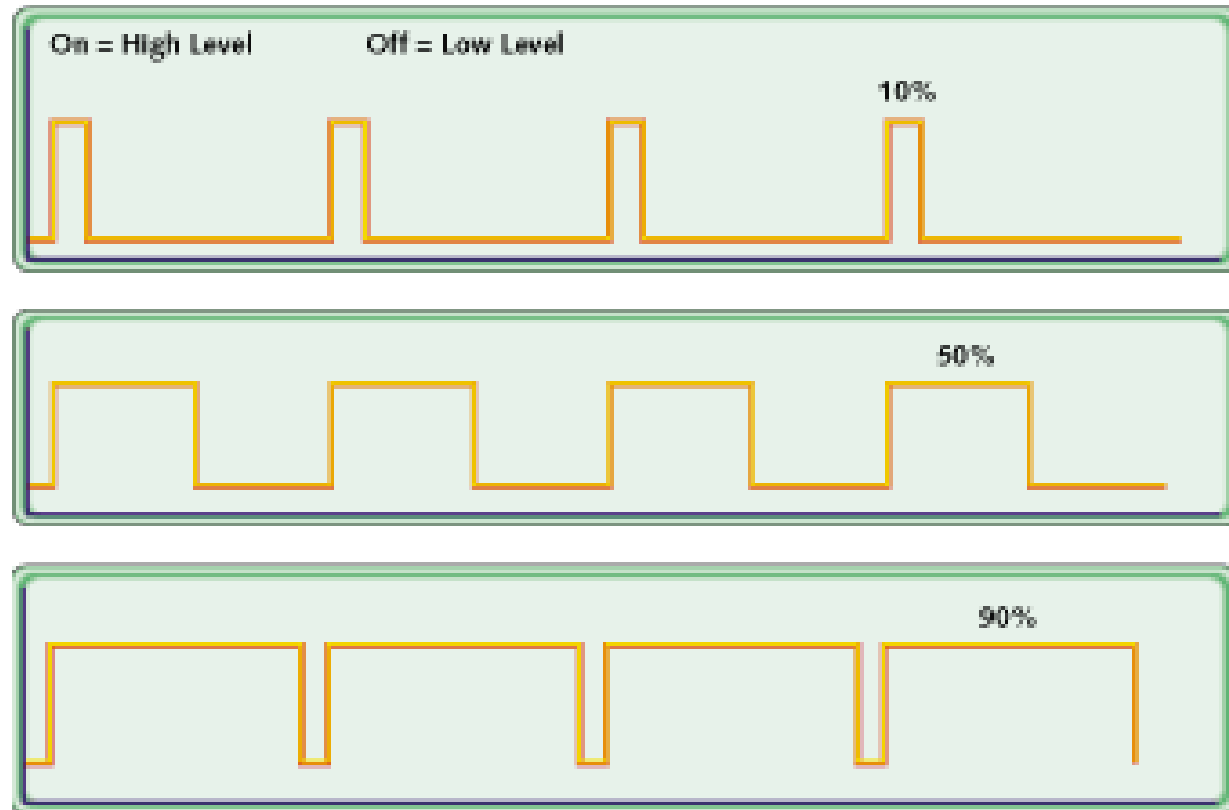
This gives high torque even at low speeds.

Used on Arduino pins 3, 5, 6, 9, 10, 11

Code:

```
analogWrite(pin,value);
```

Value 0 - 255





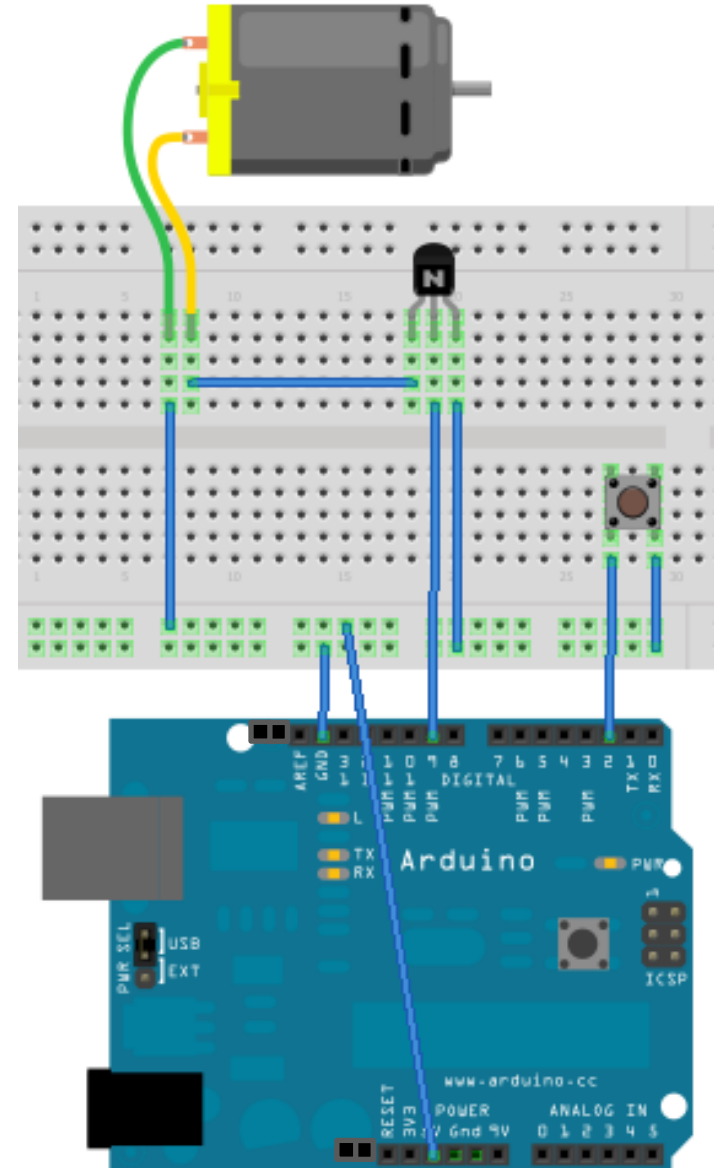


# Arduino motor control

Motor control according to switch

```
void setup() {  
  pinMode(9, OUTPUT);  
  pinMode(2, INPUT_PULLUP);  
}
```

```
void loop() {  
  if (digitalRead(2) == 0) {  
    analogWrite(9,100);  
  }  
  else {  
    analogWrite(9,0);  
  }  
}
```





# Reading analog signals

- Arduino has 6 analog inputs
- Read with:
  - `int value = analogRead(pin);`
  - Pin is 0-5 or A0-A5
  - Value from 0 to 1023 representing voltage of 0-5V



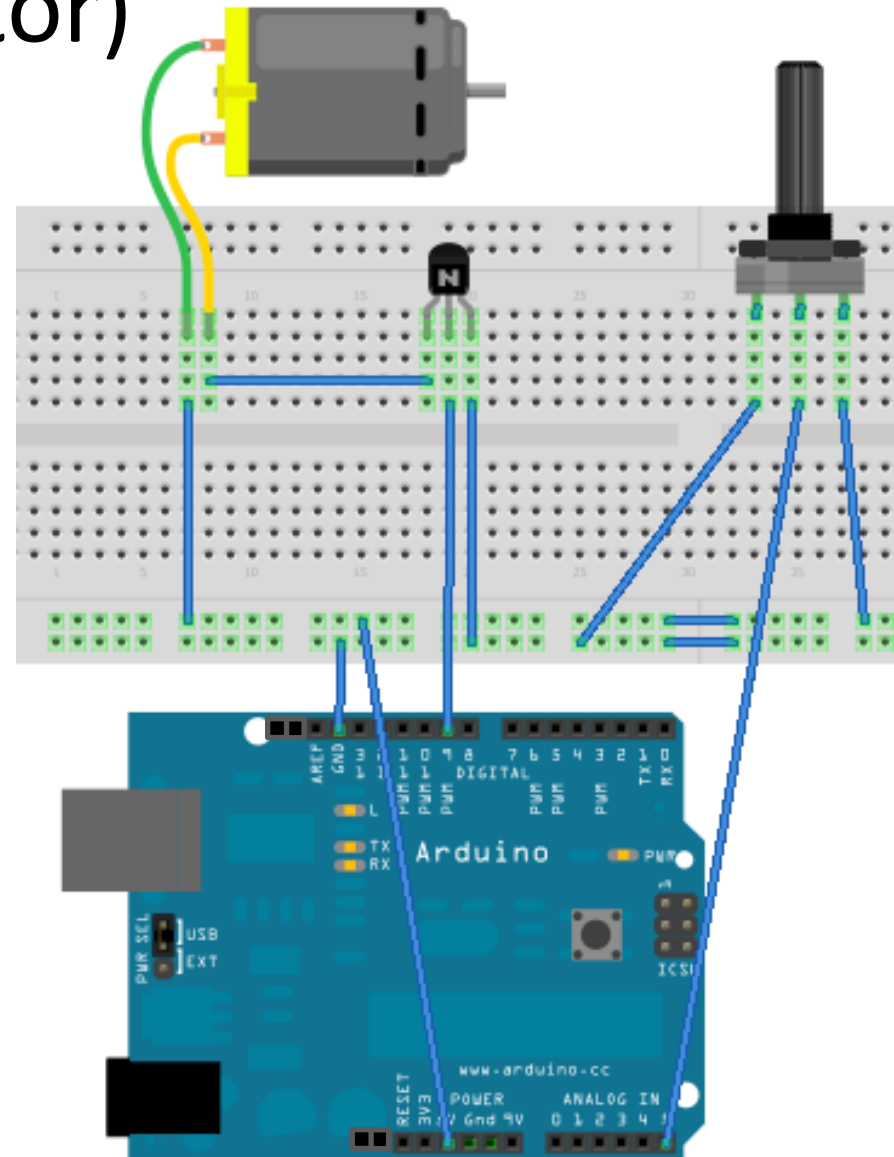
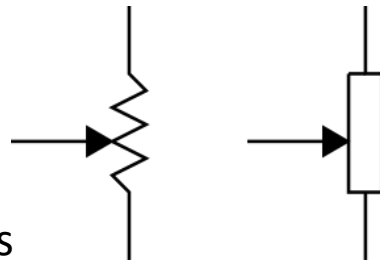
# Potentiometer (variable resistor)

Speed control using potentiometer

```
void setup() {  
  pinMode(9, OUTPUT);  
}
```

```
void loop() {  
  int light = analogRead(5);  
  analogWrite(9, light/4);  
}
```

Potentiometer  
electrical symbols



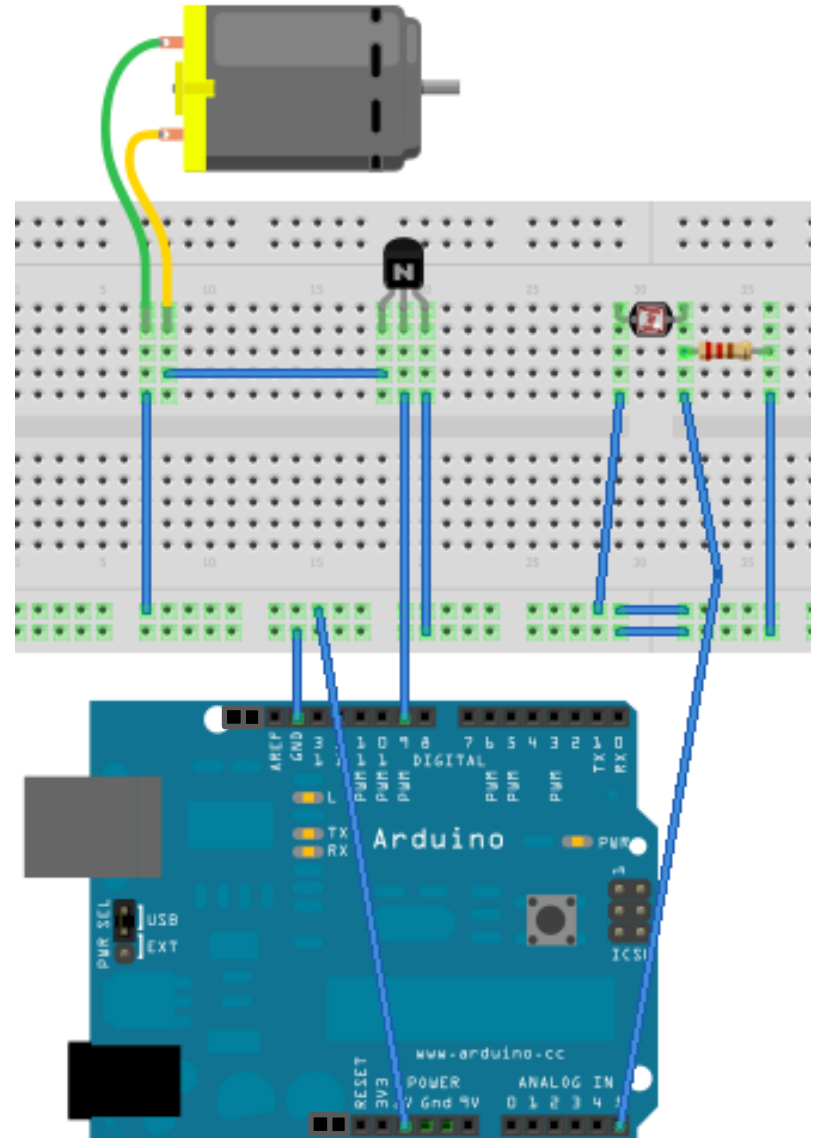
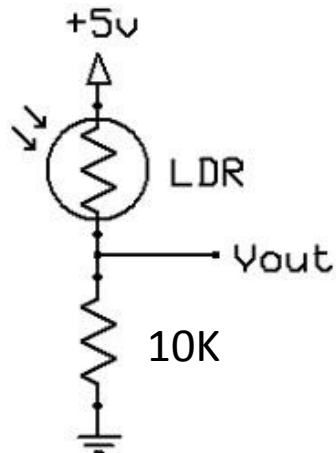


# Light dependent resistor

Speed control according to light level

```
void setup() {  
  pinMode(9, OUTPUT);  
}
```

```
void loop() {  
  int light = analogRead(5);  
  analogWrite(9, light/4);  
}
```





# Challenges!

- “Knightrider” LEDs
- Colour changing RGB LED
- Traffic lights (single set, or perhaps two sets with a pelican crossing?)
- Make some noise! (play a tune, or perhaps create a “light Theremin”? Hint: use LDR and *tone()* function)
- Line following robot
- Your own project?